## REMARKS

Claim 1 is pending in this patent application.

Claim 1 has been rejected.

Claim 1 has been cancelled.

Claims 2-21 have been added to more particularly point out and distinctly claim the Applicant's invention.

Claims 2-21 remain in the application.

The Applicant has attached an Appendix A containing the claims and the changes made to the specification for the Examiner's easy reference.

The Applicant has amended the specification to correct typographical errors. No new matter has been added as a result of the amendments to the specification. The Applicant has also concurrently submitted a document entitled "Proposed Amendment to the Drawings" to correct reference numerals in the drawings. No new matter has been added to the drawings.

On Page 2 and 3 of the June 3, 2002 Office Action the Examiner rejected Claim1 under 35 U.S.C. § 103(a) as being unpatentable over United States Patent Number 5,881,307 to *Park et al.* in view of United States Patent Number 5,095,424 to *Woffinden et al.*

In response, the Applicant has cancelled Claim 1 and added new Claims 2-21. The Applicant respectfully denies and position or averment of the Examiner that is not specifically addressed by the foregoing argument and response.

## SUMMARY

For the reasons given above, the Applicant respectfully requests reconsideration and allowance of the pending claims and that this patent application be passed to issue. If any outstanding issues remain, or if the Examiner has any suggestions for expediting allowance of this patent application, the Applicant respectfully invites the Examiner to contact the undersigned at the telephone number indicated below or at ***wmunck@davismunck.com***. The Applicant respectfully denies any position or averment of the Examiner that is not specifically addressed by the foregoing argument and response.

Respectfully submitted,

DAVIS MUNCK, P.C.

Date: Oct. 3, 2002

William A. Munck
Registration No. 39,308

Docket Clerk
P.O. Drawer 800889
Dallas, Texas 75380
Tel: (972) 628-3600
Fax: (972) 628-3616
*Email: wmunck@davismunck.com*

## APPENDIX A

## VERSION WITH MARKINGS TO SHOW CHANGES MADE

## IN THE SPECIFICATION

**Page 3, Lines 3-10 has been amended to read as follows:**

In microprocessors employing pipelined architecture, it is desirable to be in the process of executing as many instructions as possible, so that each element of the pipeline is maintained busy. However, some instructions, such as instructions that load data from external memory or [stage] store data into external memory, must generally be executed in their original sequence order, so as to avoid the external memory ever being in an incorrect state. Moreover, when such instructions refer to identical external memory locations, [where] there is no particular need to wait for the actual external memory operations to complete, as the identical data is already available for the processor to operate with.

**Page 3, Line 18 to Page 4, Line 4 has been amended to read as follows:**

Accordingly, it would be advantageous to provide a technique for operating a pipelined microprocessor more quickly, by detecting instructions that load from identical memory locations as were recently stored to, without having to actually compute the referenced external memory addresses. In a preferred embodiment, the microprocessor examines the symbolic structure of instructions as they are encountered, so as to be able to detect identical memory locations by examination of their symbolic structure. For example, instructions that store to and load from an identical offset from an identical register are determined to be referencing the identical memory [locations] location, without having to actually compute the complete physical target address.

**Page 6, Line 21 to Page 7, Line 3 has been amended to read as follows:**

The microprocessor 100 reads a sequence of instructions 151 from the instruction memory 150 using the instruction fetch stage 110 (and including any associated memory read or write elements in the microprocessor 100). In a preferred embodiment, the input instruction buffer 110 includes a plurality of instructions 151 from the instruction memory 150, but there is no particular requirement [therefore] therefor.

**Page 8, Lines 1 to 4, has been amended to read as follows:**

Similarly, an instruction that stores data to external memory has a format that refers to the specific location in external memory [from] into which to store the data. The format can similarly include a base address value and an offset address value, which are used to compute the effective reference address of the instruction 151.

**Page 10, Lines 1 to 5, has been amended to read as follows:**

Although the actual first (store) instruction 151 would be physically performed and completed by external memory, the microprocessor 100 can proceed without physically performing the second (load) instruction 151. Instead, the microprocessor 100 can use the identical data from [it's] its internal register, thus removing a relative delay in microprocessor 100 operation.

**Page 12, Lines 8 to 15, has been amended to read as follows:**

At a step 217, the bypass element 121 determines whether the operand addresses that the instructions 151 refer to include identical base address values and offset address values. If so, the bypass element 121 generates a bypass signal indicating that the instructions 151 refer to the same location in external memory. If not, the bypass element 121 does not generate a bypass signal. (In alternative embodiments, the bypass element 121 may generate an inverse bypass signal). If the bypass element 121 generates a bypass signal, the method 200 proceeds with the [step 216] step 220. If not, the method 200 proceeds with the step 221.

## IN THE CLAIMS

**Claim 1 has been cancelled.**

**New Claims 2-21 have been added as follows:**

2. [New] A pipelined microprocessor capable of detecting an instruction that loads data from a first memory location that was previously stored to without computing an external memory address of said first memory location.

3. [New] A pipelined microprocessor as claimed in Claim 2 wherein said pipelined microprocessor is capable of detecting an instruction that stores data into a second memory location that was previously read from without computing an external memory address of said second memory location.

4. [New] A pipelined microprocessor as claimed in Claim 2 wherein said pipelined microprocessor is capable of detecting instructions that load data from identical memory locations that were previously stored to without computing external memory addresses of said identical memory locations.

5. [New] A pipelined microprocessor as claimed in Claim 3 wherein said pipelined microprocessor is capable of detecting instructions that store data into identical memory locations that were previously read from without computing external memory addresses of said identical memory locations.

6. [New] A pipelined microprocessor as claimed in Claim 4 wherein said pipelined microprocessor is capable of examining symbolic structure of said instructions that load data from identical memory locations that were previously stored to, and capable of detecting said instructions that load data from identical memory locations by examining said symbolic structure.

7. [New] A pipelined microprocessor as claimed in Claim 5 wherein said pipelined microprocessor is capable of examining symbolic structure of said instructions that store data into identical memory locations that were previously read from, and capable of detecting said instructions that store data into identical memory locations by examining said symbolic structure.

8. [New] A pipelined microprocessor as claimed in Claim 6 wherein said pipelined microprocessor is capable of detecting said instructions that load data from identical memory locations by identifying an identical offset address value from an identical base address value in a register within said pipelined microprocessor.

9. [New] A pipelined microprocessor as claimed in Claim 7 wherein said pipelined microprocessor is capable of detecting said instructions that store data into identical memory locations by identifying an identical offset address value from an identical base address value in a register within said pipelined microprocessor.

10. [New] A pipelined microprocessor as claimed in Claim 6 wherein said pipelined microprocessor comprises:

an instruction decode stage capable of detecting said instructions that load data from identical memory locations by identifying an identical offset address value from an identical base address value in a register within said pipelined microprocessor; and

a bypass element capable of sending a bypass signal to an instruction execution stage of said pipelined microprocessor that indicates that said instructions refer to an identical memory location..

11. [New] A pipelined microprocessor as claimed in Claim 7 wherein said pipelined microprocessor comprises:

an instruction decode stage capable of detecting said instructions that store data into identical memory locations by identifying an identical offset address value from an identical base address value in a register within said pipelined microprocessor; and

a bypass element capable of sending a bypass signal to an instruction execution stage of said pipelined microprocessor that indicates that said instructions refer to an identical memory location.

12. [New] A method for operating a pipelined microprocessor, said method comprising the step of:

detecting in said pipelined microprocessor an instruction that loads data from a first memory location that was previously stored to without computing an external memory address of said first memory location.

13. [New] A method for operating a pipelined microprocessor as claimed in Claim 12, said method further comprising the step of:

detecting in said pipelined microprocessor an instruction that stores data into a second memory location that was previously read from without computing an external memory address of said second memory location.

14. [New] A method for operating a pipelined microprocessor as claimed in Claim 12, said method further comprising the step of:

detecting in said pipelined microprocessor instructions that load data from identical memory locations that were previously stored to without computing external memory addresses of said identical memory locations.

15. [New] A method for operating a pipelined microprocessor as claimed in Claim 13, said method further comprising the step of:

detecting in said pipelined microprocessor instructions that store data into identical memory locations that were previously read from without computing external memory addresses of said identical memory locations.

16. [New] A method for operating a pipelined microprocessor as claimed in Claim 14, said method further comprising the steps of:

examining in said pipelined microprocessor symbolic structure of said instructions that load data from identical memory locations that were previously stored to; and

detecting said instructions that load data from identical memory locations by examining said symbolic structure.

17. [New] A method for operating a pipelined microprocessor as claimed in Claim 15, said method further comprising the steps of:

examining in said pipelined microprocessor symbolic structure of said instructions that store data into identical memory locations that were previously read from; and

detecting said instructions that store data into identical memory locations by examining said symbolic structure.

18. [New] A method for operating a pipelined microprocessor as claimed in Claim 16, said method further comprising the steps of:

detecting in an instruction decode stage of said pipelined microprocessor said instructions that load data from identical memory locations by identifying an identical offset address value from an identical base address value in a register within said pipelined microprocessor; and

sending a bypass signal from a bypass element to an instruction execution stage of said pipelined microprocessor wherein said bypass signal indicates that said instructions refer to an identical memory location.

19. [New] A method for operating a pipelined microprocessor as claimed in Claim 17, said method further comprising the steps of:

detecting in an instruction decode stage of said pipelined microprocessor said instructions that store data into identical memory locations by identifying an identical offset address value from an identical base address value in a register within said pipelined microprocessor; and

sending a bypass signal from a bypass element to an instruction execution stage of said pipelined microprocessor wherein said bypass signal indicates that said instructions refer to an identical memory location.

20. [New] A method for operating a pipelined microprocessor, said method comprising the steps of:

detecting a first instruction that stores data to a first memory location, said first instruction comprising syntax for computing an effective address for said first memory location;

detecting a second instruction that loads data from a second memory location, said second instruction comprising syntax for computing an effective address for said second memory location;

determining said syntax for said first instruction and said syntax for said second instruction;

using said syntax for said first instruction and said syntax for said second instruction to determine a relationship between said first memory location and said second memory location, without computing said effective address for said first memory location and without computing said effective address for said second memory location; and

using said relationship to determine whether to perform one of said first instruction and said second instruction.

21. [New] A method for operating a pipelined microprocessor as claimed in Claim 20 wherein said syntax for said first instruction and said syntax for said second instruction refer to an identical memory location.